A Novel Mountain Driving Unity Simulated Environment for Autonomous Vehicles

Xiaohu Li , Zehong Cao * , Quan Bai
School of ICT, University of Tasmania, Australia
xiaohul@utas.edu.au, zehong.cao@utas.edu.au, quan.bai@utas.edu.au

Abstract

The simulated driving environment provides a low cost and time-saving platform to test the performance of the autonomous vehicle by linkage with existing machine learning approaches. However, most of existing simulated driving environments focus on building flat roads in urban areas. Still, they neglected to endeavour the tough steep, curvy hill roads, such as mountain paths around suburban areas. In this study, by deploying in Unity engine, we developed the first complex mountain driving simulated environment with characterizing continuous curves and up/downhill. Then, two state-of-art reinforcement learning (RL) algorithms are used to train a vehicle agent and test the performance of autonomous vehicles in our developed simulated environment. Also, we set 5 different levels of vehicle's speeds and observe the cumulative rewards during the vehicle agent training. Our demonstration presents the developed environment supports for complex mountain scenario configurations and RL-based autonomous vehicles, and our findings show that the vehicle agent could achieve high cumulative rewards during the training stage, suggesting that our work is a potential new simulation environment for autonomous vehicles research. The demonstration video can be viewed via the link: https://youtu.be/0wSqGeCn-NU.

Introduction

Since training a vehicle agent to learn driving behaviours in the real world is uneconomical and time-consuming, it is necessary to initially use a driving simulation environment to develop autonomous driving technologies (Kiran et al. 2020). The simulator builds some synthesis environment with multiple scenarios that allow us to collect a large number of training datasets from the vehicle agent without considering unsafe behaviours. The existing simulators have been widely deployed for various purposes, especially in targeting urban transportation issues, such as CARLA (Dosovitskiy et al. 2017), SUMO (Lopez et al. 2018), AIRSIM (Shah et al. 2018), and DeepDrive (Quiter and Ernst 2018), as shown in Figure 1-A. In specific, CARLA, AIRSIM, and DeepDrive simulators provide the urban environment and reproduce the low-speed, flat and wide road scenarios, which are used to evaluate the performance of traditional controlling or machine learning based-agent for motion planning and decision making. SUMO is known for simulating urban traffic environment and planning sports.

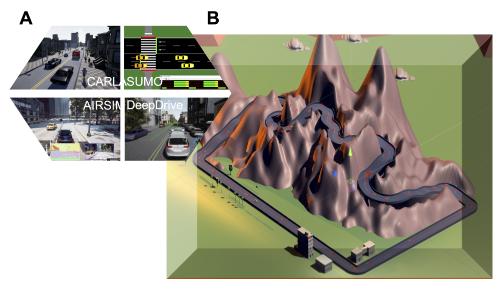

Figure 1: A: Four existing driving simulators: CARLA, SUMO, AIRSIM, and DeepDrive. B: Our developed mountain routes for autonomous vehicles in Unity.

However, the above simulators still cannot cover all of the circumstances for automatic vehicles. Thus, it is worth to develop a novel simulation environment that can generate suburban scenarios with changes in road altitudes and curvatures that have with flexibility and scalability, and meet the growing demand of machine learning approaches to support autonomous driving tasks (Osinski et al. 2019). In this study, we develop a suburban driving simulator with steep, curvy, narrow paths, such as mountain roads, which expects to simulate suburban driving environment areas that contain challenging road types, such as continuing steep, curvy hill roads.

In terms of training an autonomous vehicle, in such a complex curvy hill road driving environment that we proposed, a self-sustaining driving model may require a high driving precision with multiple perception-level tasks (Kiran et al. 2020). Traditional control approaches may be challenging to handle these circumstances, such as optimal trajectories of a vehicle, speed controls, and obstacle avoidances on a mountain road. In a high dimensional sequential decision process, reinforcement learning (RL) is recommended to be used for an agent training and building an autonomous vehicle model (Sallab et al. 2017). Thus, we assume that state of the art (SOTA) RL algorithms, such as Proximal Policy Optimization (PPO) (Schulman et al. 2017) and Soft Actor-Critic (SAC)(Haarnoja et al. 2018), could also be a promis-

*Corresponding author.

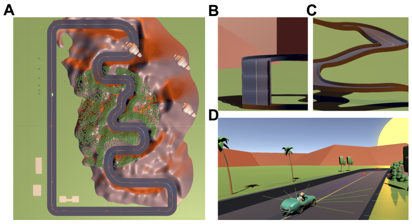

Figure 2: The road types of the mountain paths (A, B, and C) and the autonomous vehicle (D).

ing solution for autonomous driving in specific to mountain roads.

Our Proposed Mountain Driving Environment

In this study, we used the Unity engine (Buyuksalih et al. 2017) as the platform to building the environment as it has good flexibility and scalability for creating a simulated environment and provide some basic games, such as Karting that includes prototypes of essential vehicles and road models for developers. Based on the advantages of Unity, we proposed a new autonomous driving environment to simulate mountain road [1] , as shown in Fig. 1-B.

The Creation of Mountain Paths

As shown in Figure 2-A, the simulated mountain route is divided into two areas, the plain area and the mountainous area. The plain area consists of two straight roads and a 90-degree curve on the flat ground, as demonstrated in Figure 2-B. As partially presented in Figure 2-C, the mountain area consists of with straight uphill roads, “S” curve uphill roads, slightly curved roads, “S” curve downhill roads, and straight downhill roads.

The Creation of Autonomous Vehicles

As shown in Figure 2-D, a vehicle agent is created, which used 9 distance sensors to observe the surrounding obstacles and determine the direction by detecting the checkpoints. The outputs of this vehicle agent include 5 actions (left, right, straight-forward, forward acceleration, and reversing acceleration) to drive forward and avoid collisions. The acceleration speed of agent sets as 2.78 m/s^2, which is the standard acceleration speed for physical vehicles. The maximum speed of the agent set at 5 different levels (36km/h, 47km/h, 60km/h, 79km/h, and 90km/h), where 47km/h is the most commonly used speed for mountain roads.

For RL-agent training, the goal is to avoid collisions and drive in the right direction as quickly as possible. The rewards include a hit penalty of -1, a direction bonus of +1, and a speed bonus of 0.02, the agent learned that the main goal is to avoid collisions and drive in the right direction as quickly as possible. Of note, we applied in the Unity ML-Agents toolkit (Juliani et al. 2018), that allows RL algorithms to communicate with Unity simulations and evaluate the performance of the agent in a simulation environment (Cao et al. 2020).

[1]Github repository: https://github.com/Xiaohu-LeoLee/MountainRoad_AD

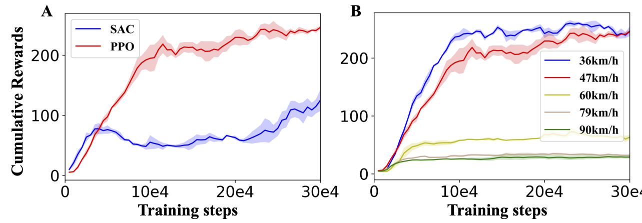

Figure 3: A: The cumulative rewards (mean ± standard deviation) of PPO and SAC training performance (random seed = 3) with the setting of of maximum speed: 47km/h. B: The cumulative rewards (mean ± standard deviation) of PPO training performance (random seed = 3) with the setting of five different levels of maximum speed: 36km/h, 47km/h, 60km/h, 79km/h, and 90km/h.

Experiments and Results

Experiment 1

Here, we expect to observe if an RL-agent can handle our developed complex suburban mountain environment. Two SOTA RL algorithms, PPO and SAC, are used to test the vehicle agent’s performance with the setting of a maximum speed 47km/h that is the most commonly used speed for mountain routes. As shown in Figure 3-A, after 50k training steps, the cumulative rewards of SAC tends to be stabilized while PPO continues to rise sharply. Both algorithms achieved the positive and coverage cumulative rewards, implying that the RL-agent allows being trained and tested in our proposed simulated mountain route environment.

Experiment 2

As we set 5 different levels of maximum speeds (36km/h, 47km/h, 60km/h, 79km/h, and 90km/h) for the PPO vehicle agent, for the second experiment, we expect to see how dynamic changes of speed ranges influencing in RL training for our proposed simulated mountain route. As shown in Figure 3-A, for the situations of the maximum speed over 60 km/h, the PPO algorithm will be challenging to handle mountain roads due to the trade-off challenge between high speeds and curve up/downhill roads.

Conclusions

In this study, we present a novel mountain driving unity simulated environment for autonomous vehicles, which makes up for the existing driving simulators only focus on urban traffic scenarios. Our developed environment is featuring with complex curvy, narrow up/downhill scenarios, along with deploying SOTA RL to train autonomous vehicles. The experiment results show that our developed environment is sufficient for autonomous vehicles training and testing, specific to complex mountain suburban areas.

References

Buyuksalih, I.; Bayburt, S.; Buyuksalih, G.; Baskaraca, A.; Karim, H.; and Rahman, A. A. 2017. 3D Modelling and Visualization Based on the Unity Game Engine–Advantages and Challenges. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4: 161.

Cao, Z.; Wong, K.; Bai, Q.; and Lin, C.-T. 2020. Hierarchical and Non-Hierarchical Multi-Agent Interactions Based on Unity Reinforcement Learning. In *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems*, 2095–2097.

Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; and Koltun, V. 2017. CARLA: An open urban driving simulator. *arXiv preprint arXiv:1711.03938* .

Haarnoja, T.; Zhou, A.; Abbeel, P.; and Levine, S. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290* .

Juliani, A.; Berges, V.-P.; Vckay, E.; Gao, Y.; Henry, H.; Mattar, M.; and Lange, D. 2018. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627* .

Kiran, B. R.; Sobh, I.; Talpaert, V.; Mannion, P.; Sallab, A. A. A.; Yogamani, S.; and Pérez, P. 2020. Deep reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2002.00444* .

Lopez, P. A.; Behrisch, M.; Bieker-Walz, L.; Erdmann, J.; Flötteröd, Y.-P.; Hilbrich, R.; Lücken, L.; Rummel, J.; Wagner, P.; and WieBner, E. 2018. Microscopic traffic simulation using sumo. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2575–2582. IEEE.

Osinski, B.; Jakubowski, A.; Milos, P.; Zikcina, P.; Galias, C.; and Michalewski, H. 2019. Simulation-based reinforcement learning for real-world autonomous driving. *arXiv preprint arXiv:1911.12905* .

Quiter, C.; and Ernst, M. 2018. deepdrive/deepdrive: 2.0. doi:10.5281/zenodo.1248998. URL https://doi.org/10.5281/zenodo.1248998.

Sallab, A. E.; Abdou, M.; Perot, E.; and Yogamani, S. 2017. Deep reinforcement learning framework for autonomous driving. *Electronic Imaging* 2017(19): 70–76.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* .

Shah, S.; Dey, D.; Lovett, C.; and Kapoor, A. 2018. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and service robotics*, 621–635. Springer.